UNIVERSITÉ
Concordia
UNIVERSITY

COMP 478/6771 Image Processing

Adaptive Logarithmic Mapping for displaying

High Contrast Scenes

Submitted by:

Name: Dhruv Goyani
Student ID: 40121588

Submitted to:

Prof. Yiming Xiao

Motivation and Contributions:

In the modern digital world, the image capturing technology is advancing at a fast pace and moving towards great contrast representation[1]. Along with that, many imaging and rendering software needs accurate information of luminance in the form of High Dynamic Range textures to capture precise appearance of a scene. The main motivation for using adaptive logarithmic mapping to display high contrast scenes was to address the need for a quicker algorithm, which could automatically generate high quality images which have high dynamic range of luminance. The figure below shows the flowchart for the algorithm in [1].

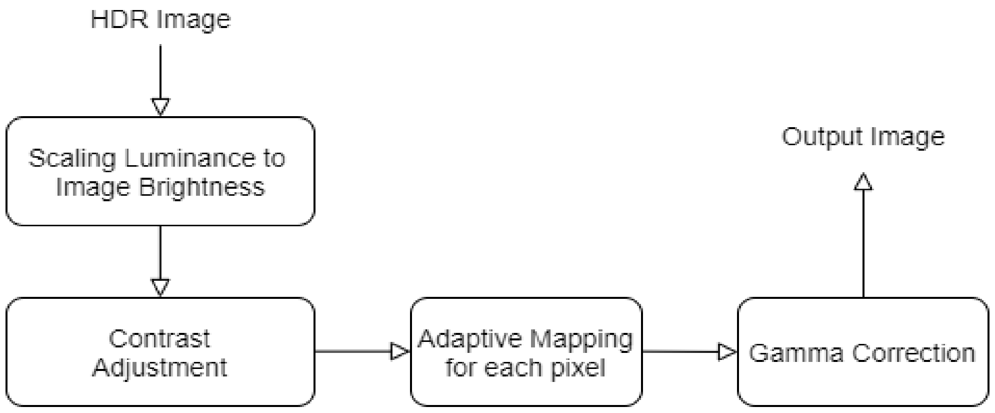

HDR Image
Scaling Luminance to Image Brightness
Contrast Adjustment
Adaptive Mapping for each pixel
Gamma Correction
Output Image

Figure 1:Flowchart for Adaptive Logarithmic Mapping for Displaying High Contrast Scenes

The algorithm was implemented by using Tone mapping based on the Stockham Logarithmic relation. To find the initial scale factor to output the image brightness, appropriate scaling of the luminance to the brightness of the image was performed using similar approach to the *Tumblin and Rushmeier* by calculating the logarithmic average of a scene based on luminance values for all the pixels. Furthermore, after applying the contrast adjustment on the obtained image, because of the non-linearity of displaying devices, gamma correction had to be applied. The final output contained fine details of the scene. While comparing the above implemented algorithm with [2], which was based on two-scale decomposition ((i) *base layer* which consists of large-scale features and a (ii) *detail layer*), it only reduces the contrast of the base layer thereby preserving the details. The focus of the paper was to develop a fast and robust edge-preserving bilateral filter that blurs the small variations but preserves the edges. In [1] paper, it used *'global tone mapping'* in which a single mapping function is used for all pixels in the given image. While the [2] algorithm uses *'local tone mapping'* the major drawback of that is the presence of haloing artifacts. The contributions of [2] are Bilateral Filtering and Robust Statistics, fast bilateral filtering, uncertainty of the output filter, and Contrast Reduction, which is fast, stable and requires no parameter setting. Comparing the results, the [2] algorithm produces better results than [1] because it uses local tone mapping, and it preserves the details while reducing the contrast, rather than using the same mapping function (global tone mapping in [1]), that does not directly address contrast reduction. While [2] gives better results, it also takes more computational time to run compared to [1], because of local tone mapping.

References

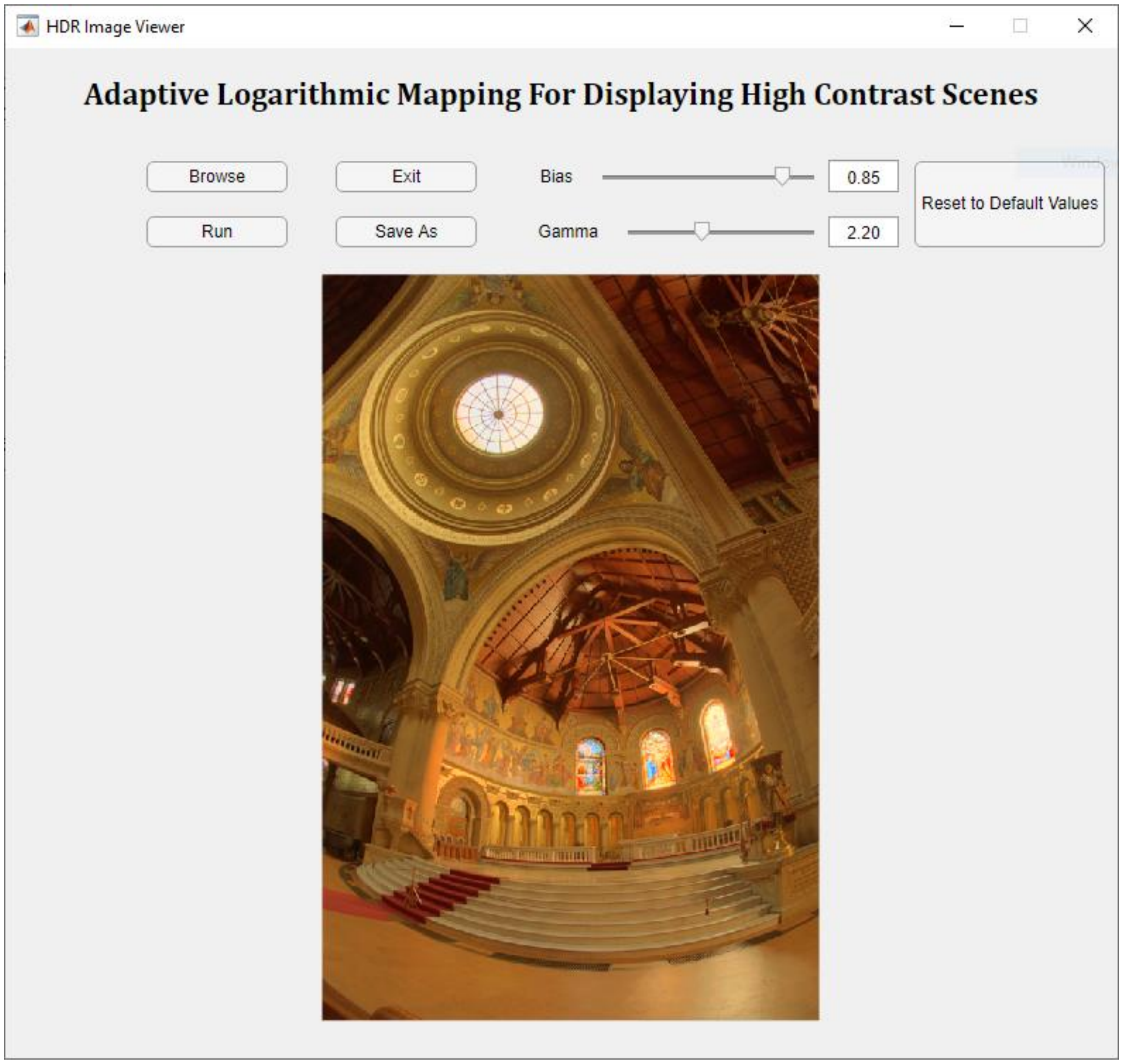
[1] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, “Adaptive Logarithmic Mapping For Displaying High Contrast Scenes,” *Computer Graphics Forum*, vol. 22, no. 3, pp. 419–426, 2003.

[2] F. Durand and J. Dorsey, “Fast bilateral filtering for the display of high-dynamic-range images,” *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 257–266, 2002.

Implementation of Algorithm:

I implemented the algorithm as presented in the paper and tried to achieve almost equivalent results according to the paper. I used multiple images (.hdr format) for testing the output.

There were some assumptions in the original proposed method. First assumption was the 'exposure factor'. It was mentioned in the paper that the exposure factor is optional by which Luminance values are scaled. I assumed the exposure factor as 1 in my implementation. The value of 'maximum luminance capacity' L_{dmax} is set to 100 cd/m^2 which is a common reference value for CRT displays. Also, the value of 'maximum luminance capacity' L_{wmax} was assumed to be 230 cd/m^2. The bias parameter was selected as b = 0.85 by averaging preferences and realism. In my implementation, I have provided a variable bias value ranging from 0 to 1 to let the user observe the output for different bias values.


HDR Image Viewer
Adaptive Logarithmic Mapping For Displaying High Contrast Scenes
Browse
Exit
Run
Save As
Reset to Default Values

Bias 0.85
Gamma 2.20

Figure 2: The User Interface

The below figures are the output images of *'memorial.hdr'* for different bias values:

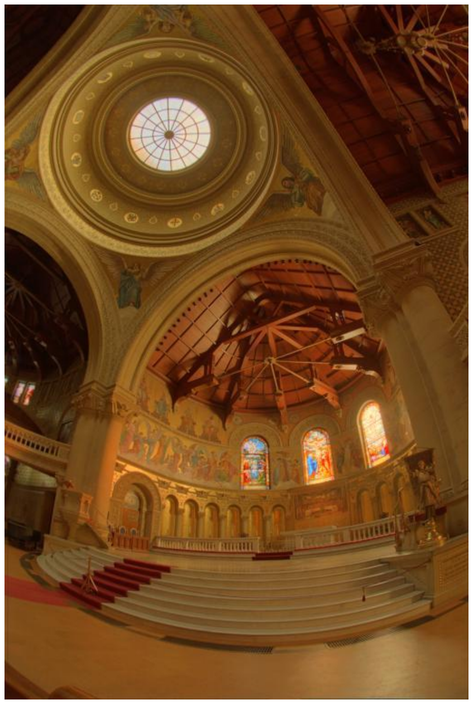


Figure 3: Memorial Image for b=1

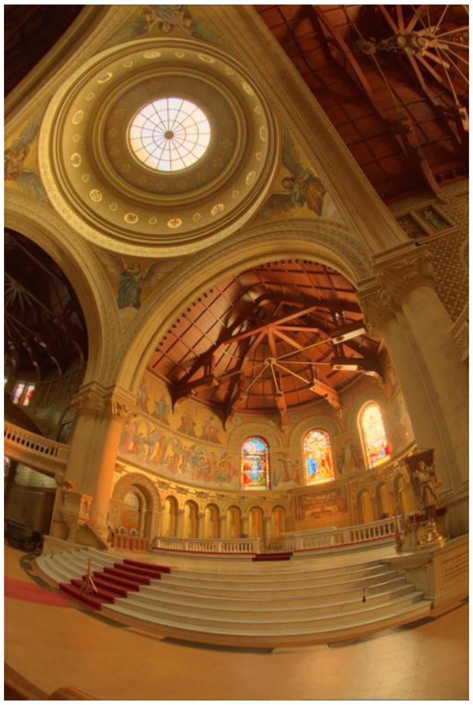


Figure 4:Memorial Image for b=0.85

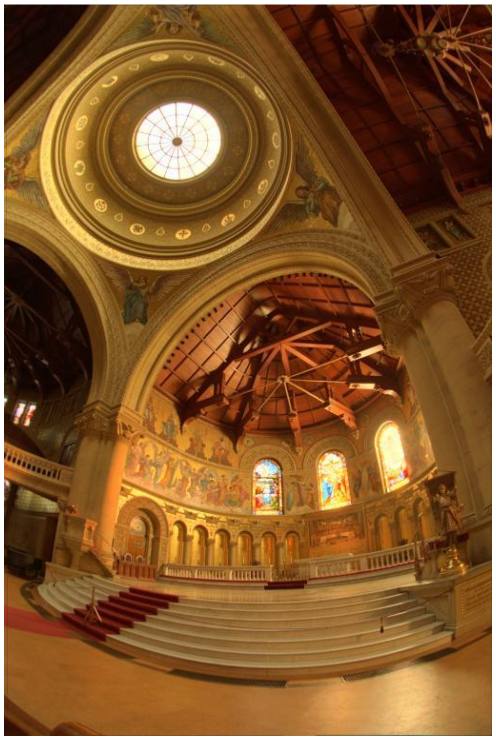


Figure 5: Memorial Image for b=0.7

Stanford Memorial Church, Copyright Paul Debevec. Dynamic Range = 343112:1

It was mentioned in the paper that the brightness is approximately doubled for b=0.85 and tripled for b=0.7 in respect to b=1.0. The same results can be observed in above 3 images (observe the top circular dome from where the light rays are passing).

In the gamma correction process, the standard value of gamma = 2.2 was assumed. I also provided the functionality to the user for changing the gamma value between 1 to 4 and observing the effect of gamma correction on the images. I used default gamma correction process rather than using the one which was provided in the paper i.e., based on ITU-R BT.709 standard.

These are the few output images on which I performed the given algorithm:

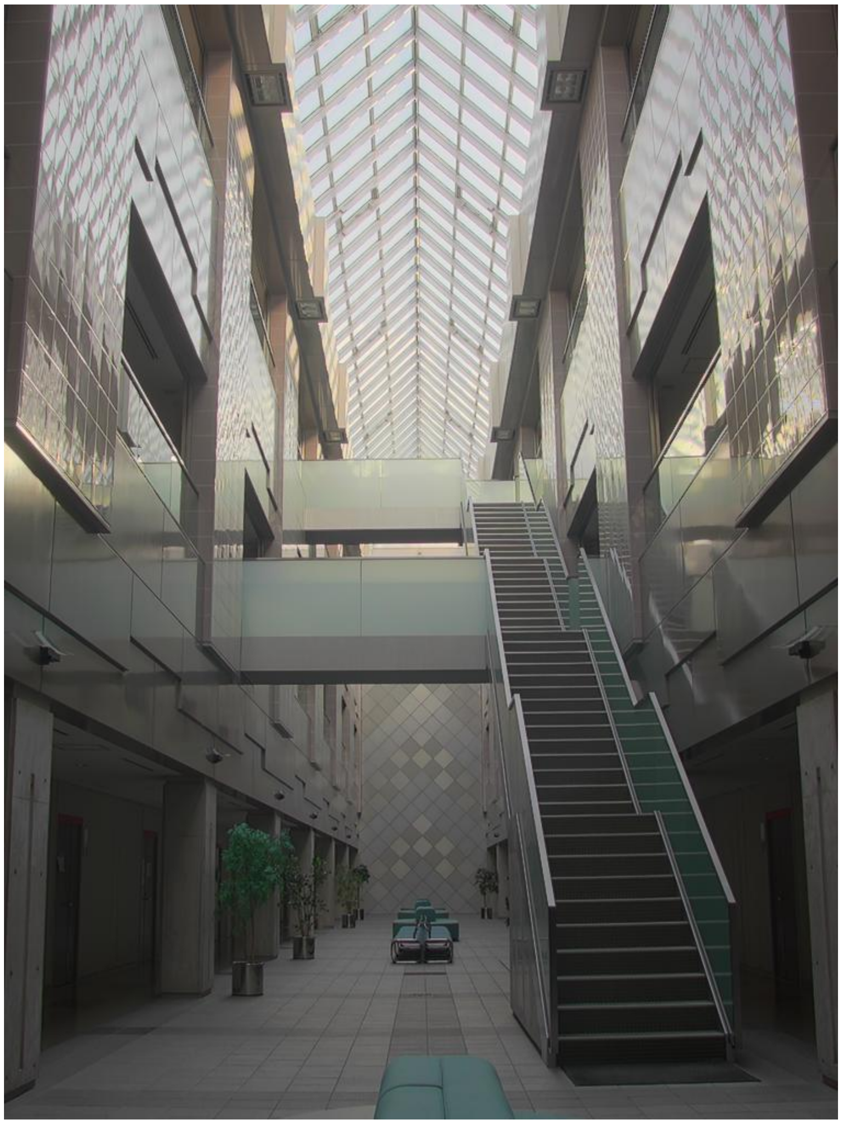


Figure 6: Atrium Morning Output Image with Default Values. Atrium Morning, Aizu University. Copyright Frederic Drago. Dynamic Range = 11751307:1

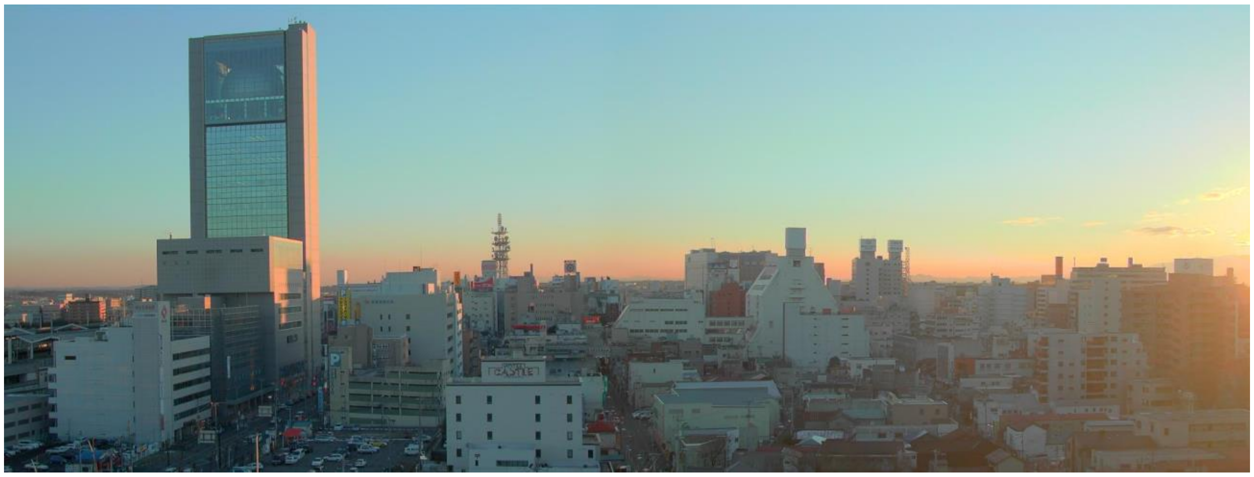


Figure 7:Buildings Output Image with Default Values. Copyright © by Frédéric Drago

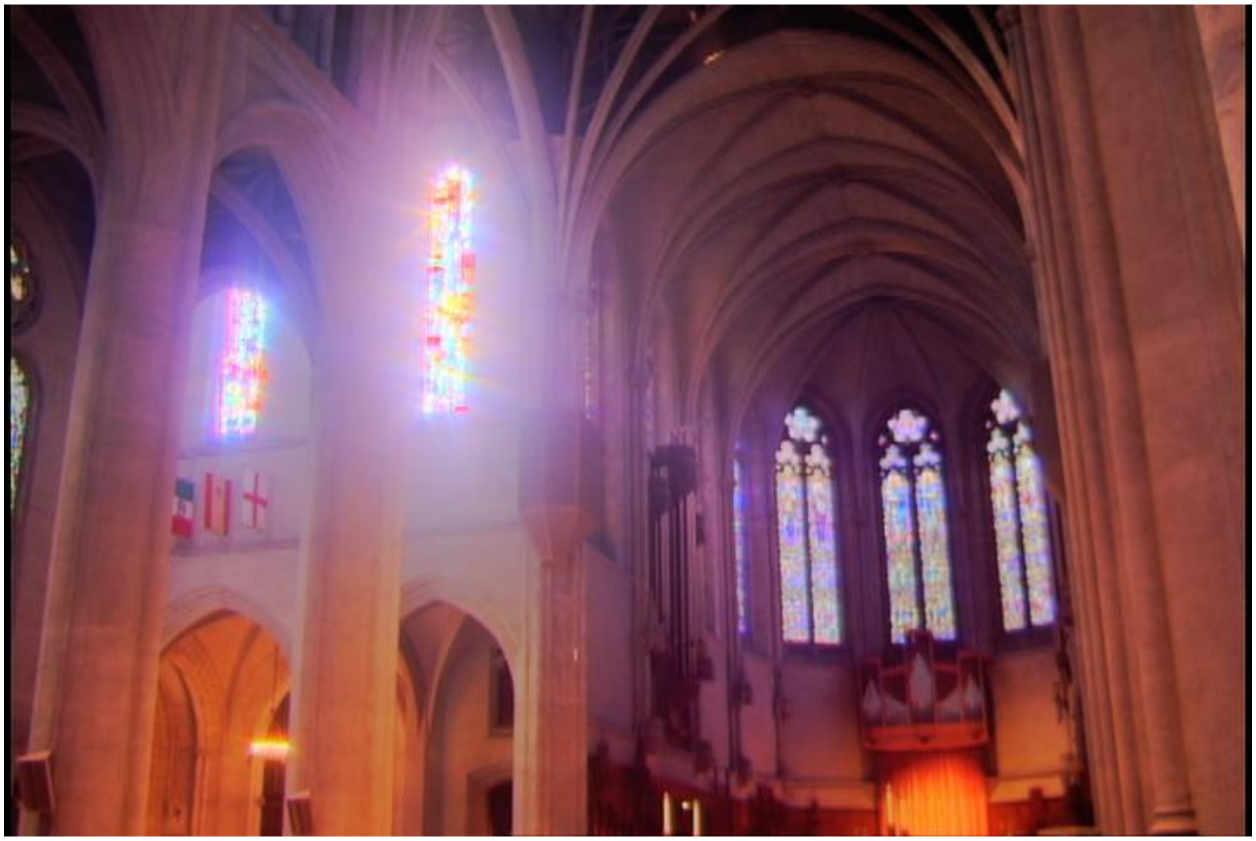


Figure 8: Nave Output. Image courtesy of Paul Debevec

I also compared the running time for each image and compared with the time given in the paper. Relatively, the running time is almost same for each image. The table below shows the comparison of running times for the images:

Image	**Size (Pixels)**	**My Time (Base) in seconds**	**Author's Time (Base) in seconds**
MEMORIAL	512x768	0.133786	0.143
NAVE	720x480	0.110722	0.126
ATRIUM	1016x760	0.241322	0.281

Table 1: *Comparison of my Tone Mapping Execution time with author's time.*

In the time mentioned above, the tone mapping routine is taken into consideration. Image IO, color space transformation, and the initial calculation of L_{wa} remains constant and is not a part of the table. Very small difference is seen because of the different specifications of compiler and machine. But relatively it can be seen that the time is approximately same.

I have provided extra functionalities such as saving of the image and loading HDR image, so the user can save the image and load any HDR image to observe various effects on output image.

The method that I used during implementation are the basic methods that the algorithm is based upon. While reimplementing the algorithm, there were 2 places where I faced difficulties. One is the optimization part. The original tone mapping is pixelwise operation. So, the computation time increases linearly with the number of pixels, which is slow for real time applications. The author optimized the above traditional method by observing that if luminance difference among pixels is below a certain threshold, the bias function can be evaluated just once for averaging luminance value. The threshold was not mentioned clearly in the paper. Furthermore, they used Pade's approximation of $\log(x+1)$ for low radiance values to speed up the execution time. I did not implement the HDR Movie player where the author compares the FPS. Another difficulty that I faced was the implementation of Gamma transfer function(i.e., according to ITU-R BT.709 standard). Instead of that, I used simple Gamma function for mapping discussed in class.

The advantage of the method used in this algorithm is that it is uses global tone mapping, which executes faster, but on the other end, its disadvantage is that by using it, contrast reduction occurs. It does not address the contrast reduction. Another disadvantage of using it is the assumption of bias value.

As a future work, automatic detection of bias value as a function of the scene content can be performed.